IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Raymond E. SUORSA et al.	§	Confirmation No.:	9523
Serial No.:	09/838,142	§	Group Art Unit:	2152
Filed:	April 20, 2001	§	Examiner:	V. D. Lesniewski
For:	Automated Provisioning of Computer Networks Using a Network Database Data Model	§	Docket No.:	200704486-1

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: January 8, 2007

Sir:

Appellants hereby submit this Appeal Brief in connection with the above-identified application. A Notice of Appeal was filed on November 8, 2007.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST 3
II. RELATED APPEALS AND INTERFERENCES 4
III. STATUS OF THE CLAIMS 5
IV. STATUS OF THE AMENDMENTS 6
V. SUMMARY OF THE CLAIMED SUBJECT MATTER 7
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 9
VII. ARGUMENT 10
A. Overview of Suzuki 10
B. Claims 32-33 and 36-38 10
1. The "Database" Limitation 10
2. The "Queue" Limitation 12
3. The "Placing the Queue in a Reboot Status" Limitation 13
C. Claims 34-35 15
VIII. CONCLUSION 16
IX. CLAIMS APPENDIX 17
X. EVIDENCE APPENDIX 19
XI. RELATED PROCEEDINGS APPENDIX 26

I. REAL PARTY IN INTEREST

The real party in interest is the Hewlett-Packard Development Company (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas. HPDC is a wholly owned affiliate of Hewlett-Packard Company (HPC). The rights in the subject application were previously owned by Opsware, Inc., formerly known as Loudcloud, Inc.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

Originally filed claims:	1-21.
Added claims:	22-38.
Claim cancellations:	1-31.
Presently pending claims:	32-38.
Presently appealed claims:	32-38.
Presently allowed claims:	None.
Presently objected claims:	None.

IV. STATUS OF THE AMENDMENTS

There were no amendments filed subsequent to the final Office Action of June 5, 2007 (hereinafter "Final Office Action").

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

Computer systems that are used to provide the wide variety of services available on the Internet include a collection of software applications that operate to provide such services, such as, for example, web servers that provide stored informational content associated with a particular Internet website. Specification of the subject application (hereinafter "Specification"), pp. 1-2, lines 1-16 of ¶ [0003]. As the number and complexity of the services required of such systems increase, so does the number and complexity of the software applications (as well as the servers executing the software applications) used to provide the required services. Specification, p. 2, lines 16-21 of ¶ [0003]. Automating the Installation or "provisioning" of a computer system, during the operation of the system, with the software needed to provide the required services is the subject of Appellants' technological contribution.

In accordance with the invention of independent claim 32, for example, a method for executing commands (in a system having a database, a plurality of devices remote from the database and a gateway that provides a communications interface between the remote devices and the database (Specification, p. 16, lines 1-27 of ¶ [0047] and Fig. 7)) that includes storing a queue in the database containing a sequence of commands to be executed (Specification, pp. 23-24, lines 3-9 of ¶ [0063]); retrieving, at the gateway, a command from the queue and transmitting the retrieved command from the gateway to an agent running on at least one of the remote devices, for execution on the one device (Specification, pp. 24-25, lines 1-4 of ¶ [0066] and Fig. 10); and, at the gateway, receiving a message from the agent reporting the results of the execution of the command (Specification, p. 25, lines 4-5 of ¶ [0066] and Fig. 10). The method also includes retrieving, at the gateway, the next command from the queue in response to receipt of the message, and transmitting the retrieved next command to the agent for execution (Specification, p. 25, lines 10-13 of ¶ [0066] and Fig. 10); in response to receiving a message at the gateway from the agent reporting the results of the

execution of at least one command, transmitting a command from the gateway to the agent on the remote device to initiate a reboot process (Specification, p. 25, lines 1-2 of ¶ [0067] and Fig. 10); and placing the queue in a reboot status in response to the initiation of the reboot process (Specification, p. 25, lines 3-4 of ¶ [0067] and Fig. 10). The method further includes retrieving at the gateway a message from the agent indicating the completion of the reboot process at the remote device (Specification, p. 25, lines 4-6 of ¶ [0067] and Fig. 10); removing the queue from reboot status in response to the message, and checking at the gateway whether any commands remain in the queue that have not yet been completed (Specification, p. 25, lines 6-8 of ¶ [0067] and Fig. 10); and resuming the step of retrieving commands in the queue and transmitting them to the agent if uncompleted commands are determined to be present in the queue (Specification, p. 25, lines 6-8 of ¶ [0067] and Fig. 10).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 32-33 and 36-38 are unpatentable over Suzuki et al. (U.S. Pat. No. 6,816,964, hereinafter "Suzuki").

Whether claims 34-35 are unpatentable over Suzuki in view of Gonda et al. (U.S. Pat. No. 6,662,221, hereinafter "Gonda").

VII. ARGUMENT

A. Overview of Suzuki

Suzuki is directed to an agent that is downloaded from a server to a client before installation. The agent performs the installation of an install file into the client according to an execution script, and by referring to a managing record file. The agent also updates the managing record file according to the install execution state of the client. After a reboot of the client, the agent continues the installation of the install file, according to the execution script, by referring to the managing record file. Suzuki, Abstract. The execution script is part of a script or control file that is stored on a hard disk drive (HDD) within the server. Suzuki, col. 6, lines 44-45 and lines 51-56. The execution script includes definition information that determines execution or setup commands that are necessary for the installation of the install files into the client. Suzuki, col. 6, lines 60-62. The execution script is read and executed by the agent in the client. Suzuki, col. 6, lines 62-63. The agent is a program that executes processes according to the execution commands within the execution script. Suzuki, col. 6, line 66 through col. 7, line 2. Appellants respectfully note that Suzuki makes no mention at all of either a database or a queue.

B. Claims 32-33 and 36-38

1. The "Database" Limitation

In the Final Office Action, the Examiner re-asserted the rejection of claims 32-33 and 36-38 as allegedly anticipated by Suzuki, stating with regard to independent claim 32, among other things, that Suzuki disclosed, "A method for executing commands in a system having a database (figure 2, item 105)...." Final Office Action, p. 2, ¶ 8. The Examiner further asserted that,

> The previous line citations, figure 2, item 105 and figure 1, item 11, show a hard disk drive for data storage and a script file comprising execution script S. Concerning a database as being composed of records, etc., see also column 6, lines 51-59. Concerning the term queue, it is noted that the claim defines the term as "containing a sequence of commands to be executed." Again, Suzuki's script file comprises execution script S.

Final Office Action, p. 8, ¶ 15. Appellants respectfully note that the text cited by the Examiner merely describes what was already described by the Examiner: a hard disk drive that operates to store the execution script on the server. Appellants respectfully submit that this description does not support the Examiner's assertion that the Suzuki teaches "a system having a database."

Appellants respectfully note that in order for independent claim 32 to be given its broadest reasonable interpretation that is consistent with the specification, "database" must be given its plain meaning unless Appellants have expressly provided a clear definition in the specification, showing that Appellants intended to act as their own lexicographer. The ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention. "Database" is a term of art in the computer industry. The meaning that the term "database" would have to a person of ordinary skill in the computer art at the time of the invention is shown by the definition of "database" provided by the Microsoft Computer Dictionary[1], 4th Edition, 1999 (hereinafter "MS Dictionary"), namely, "[a] file composed of records, each containing fields together with a set of operations for searching, sorting, recombining, and other functions." MS Dictionary, p. 123. A "file" is further defined as, "[a] complete, named collection of information, such as a program, a set of data used by a program, or a user-created document." MS Dictionary, p. 183. Both of these definitions have previously been presented and discussed in Appellants' prior response of April 2, 2007 (hereinafter "Prior Response").

In contrast, a hard disk is defined as,"[a] device containing one or more inflexible platters coated with material in which data can be recorded magnetically, together with their read/write heads, the head-positioning mechanism, and the spindle motor in a sealed case that protects against outside contaminants." MS Dictionary, p. 212 (also previously presented and

[1] Appellants have provided copies of the relevant sections of the cited reference in the Evidence Appendix.

discussed in Appellants' Prior Response). Appellants respectfully submit that since a database is defined as a type of file, and a device containing inflexible platters and other mechanical components is not a file, it should be clear that the broadest reasonable interpretation of the term "database" does not include a hard disk drive, as asserted by the Examiner.

While Appellants acknowledge that a hard disk drive may be used to store a database, it does not automatically follow that every hard disk drive must include a database. Further, the Examiner has not expressly presented the argument that hard disk drive 105 of Suzuki inherently includes a database. "The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic." MPEP § 2112 (citing *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993)). Further, "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." MPEP § 2112 (citing *Ex parte Levy,* 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990)). Since the hard disk drive 105 of Suzuki is not expressly described as a database, and the Examiner has not provided a basis in fact and/or technical reasoning to reasonably support the determination that the hard disk drive 105 inherently includes a database, Appellants respectfully submit that the cited reference fails to teach all of the limitations of independent claims 32, and thus, for at least these reasons, does not anticipate the claim under 35 U.S.C. § 102(e).

2. The "Queue" Limitation

Appellants further respectfully note that the Examiner appears to have interpreted the script file 11 of Suzuki as reading on a queue containing a sequence of command to be executed. Appellants respectfully traverse the Examiner's characterization of the term "queue," noting that this term is also a term of art in the computer industry with a known meaning to one of ordinary skill in the computer art at the time of the invention. The term "queue" would

have been so known to one of ordinary skill as "[a] multi-element data structure from which (by strict definition) elements can be removed only in the same order in which they were inserted; that is, it follows a first in, first out (FIFO) constraint. There are also several types of queues in which removal is based on factors other than order of insertion–for example, some priority value assigned to each element." MS Dictionary, p. 368 (also previously presented and discussed in Appellants' Prior Response).

In contrast to the known definition of "queue," the script file of Suzuki is described as "a control file, storing therein an execution script S, i.e. execution control information, which prescribes an execution process of installation into the client 200...." Suzuki, col. 6, lines 53-56. Since an execution script is not a multi-element data structure from which elements are removed according to the order inserted (or another factor), the broadest reasonable interpretation of "queue" would not included the execution script of Suzuki.

Regarding the Examiner's assertion (concerning the term queue) that "the claim defines the term as 'containing a sequence of commands to be executed,' " Appellants further respectfully note that independent claim 32 requires a database "containing a sequence of commands to be executed," which is a description of the content of the database. Appellants respectfully submit that the claim limitation does not serve to re-define the term "database", but merely to describe how the database is being used within the context of the claimed invention.

For at least these reasons, Appellants respectfully submit that the Examiner has failed to support his assertion that Suzuki teaches or even suggest the queue required by independent claim 32, and thus fails to teach all of the limitations of the claim. Appellants thus respectfully submit that for at least these reasons, independent claim 32 is not anticipated by Suzuki under 35 U.S.C. § 102(e).

3. The "Placing the Queue in a Reboot Status" Limitation

The Examiner further stated that Suzuki teaches "placing the queue in a reboot status in response to the initiation of the reboot process (column 8, lines

40-45)..." Final Office Action, p. 3, ¶ 8. Appellants respectfully note that independent claim 32 requires placing the queue in a reboot status in response to the initiation of a reboot process. By contrast, the cited portion of Suzuki provides:

> A reboot is requested during installation of the operating system. The agent 12 notifies the install execution state up to then to the manager 14 in the server 100. In response to this notification, the manager 14 records it into the managing record file 10 for updating. Simultaneously, the manager 14 sets the boot flag of the client 200 to 1 (local boot) for switching from the remote boot to the local boot (step S6).

Suzuki, col. 8, lines 38-45. Appellants respectfully not that the cited text does not even mention the script file 11, which the Examiner alleges reads on the queue of independent claim 32, let alone placing the script file 11 into a reboot status.

The Examiner further asserted that the cited text "shows that a boot flag is set and a reboot is then executed. During reboot, execution commands in the script file are held and this is considered to meet the limitation of 'a reboot status.' " Final Office Action, p. 9, ¶ 18. Appellants respectfully traverse the Examiner characterization of the cited art, noting that Suzuki instead teaches, "Upon such switching, setting of a boot flag provided in the managing record file 10 corresponding to each of the clients 200a to 200n is implemented, wherein the boot flag is set to 1 in case of the local boot and 0 in case of the remote boot." Suzuki, col. 7, lines 31-35. Thus the boot flag reflects the local or remote boot state of the client, and as already stated is not related to script file 11, or any other element of the server taught by Suzuki.

For at least these reasons, Appellants respectfully submit that the Examiner has failed to support his assertion that Suzuki teaches or even suggest placing the queue in a reboot status in response to the initiation of the reboot process, as required by independent claim 32, and thus fails to teach all of the limitations of the claim. Appellants thus respectfully submit that for at

least these reasons, independent claim 32 is not anticipated by Suzuki under 35 U.S.C. § 102(e).

For at least all of the reasons described above, Appellants respectfully submit that the Examiner erred in rejecting independent claim 32 as anticipated by Suzuki under 35 U.S.C. § 102(e), and therefore further respectfully submit that independent claim 32, as well as those claims that depend upon it, are all in condition for allowance.

C. Claims 34-35

Regarding the rejection of dependent claims 34 and 35 as allegedly obvious over Suzuki in view of Gonda, Appellants respectfully note that because these claims each include all of the limitations of independent claim 32, and because none of the cited art, either alone or in conjunction with each other, teaches or even suggest all of the limitations of claim 32 for at least the reasons presented above, dependent claims 34 and 35 are not rendered obvious over the cited art under 35 U.S.C. § 103(a).

VIII. CONCLUSION

For the reasons stated above, Appellants respectfully submit that the Examiner erred in rejecting claims 32-38, and respectfully requests reversal of the rejections. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/Roberto de Leon/
Roberto de Leon
PTO Reg. No. 58,967
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANTS

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

IX. CLAIMS APPENDIX

Claims on Appeal:

32. (Previously presented) A method for executing commands in a system having a database, a plurality of devices remote from the database and a gateway that provides a communications interface between said remote devices and said database, comprising the following steps:

storing a queue in said database containing a sequence of commands to be executed;

retrieving, at said gateway, a command from the queue and transmitting the retrieved command from the gateway to an agent running on at least one of said remote devices, for execution on said one device;

at said gateway, receiving a message from the agent reporting the results of the execution of the command;

retrieving, at said gateway, the next command from the queue in response to receipt of said message, and transmitting said retrieved next command to the agent for execution;

in response to receiving a message at the gateway from the agent reporting the results of the execution of at least one command, transmitting a command from the gateway to the agent on the remote device to initiate a reboot process;

placing the queue in a reboot status in response to the initiation of the reboot process;

retrieving at the gateway a message from the agent indicating the completion of the reboot process at the remote device;

removing the queue from reboot status in response to said message, and checking at the gateway whether any commands remain in the queue that have not yet been completed; and

resuming the step of retrieving commands in the queue and transmitting them to the agent if uncompleted commands are determined to be present in the queue.

33. (Previously presented) The method of claim 32, wherein said queue is placed in said reboot status in response to receipt at said gateway of a message from the agent on the remote device indicating that the reboot process is in progress.

34. (Previously presented) The method of claim 33, wherein said agent opens a new communication session with said gateway to transmit said message.

35. (Previously presented) The method of claim 34, wherein said new communication session comprises a secure socket.

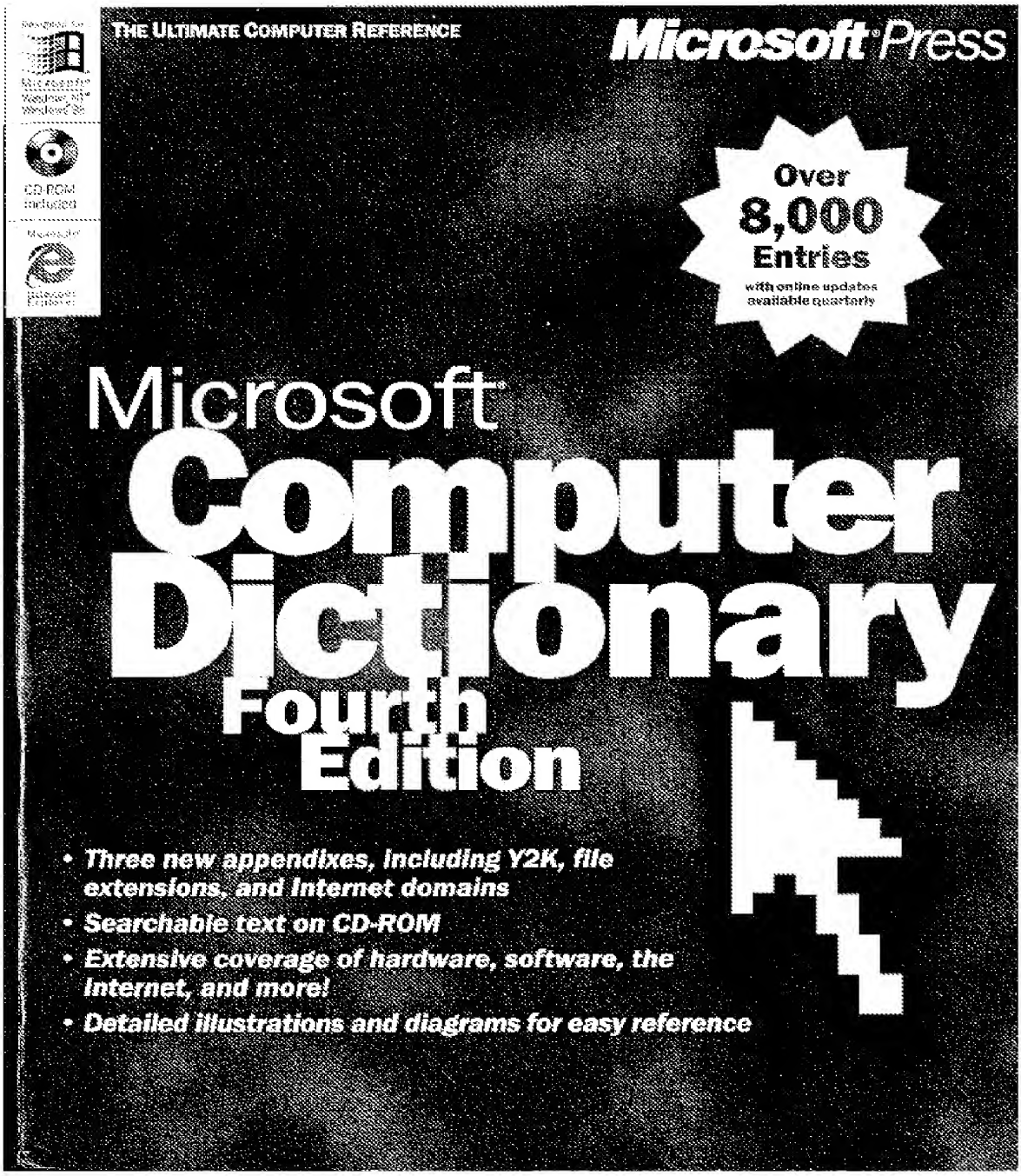
36. (Previously presented) The method of claim 32, further including the step of updating the status of the queue to indicate the command that has been most recently transmitted to the agent for execution.

37. (Previously presented) The method of claim 32, wherein said sequence of commands cause the agent to install and configure software on the remote device.

38. (Previously presented) The method of claim 32, wherein said message indicating the completion of the reboot process at the remote device includes a report of the configuration of the remote device.

X. EVIDENCE APPENDIX

Appellants have included excerpts from the Microsoft Computer Dictionary, 4th Edition, 1999, in this Appendix in support of arguments presented as to how one of ordinary in the art at the time of the invention would have understood the terms used in the claims of the subject application, as discussed in the present Appeal Brief.


THE ULTIMATE COMPUTER REFERENCE
Microsoft Press
CD-ROM included
Over 8,000 Entries
with online updates available quarterly
Microsoft
Computer Dictionary
Fourth Edition
• Three new appendixes, including Y2K, file extensions, and Internet domains
• Searchable text on CD-ROM
• Extensive coverage of hardware, software, the Internet, and more!
• Detailed illustrations and diagrams for easy reference

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 1999 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary. -- 4th ed.
p. cm.
Previous eds. published under title: Microsoft Press computer dictionary
ISBN 0-7356-0615-3
1. Computers Dictionaries. 2. Microcomputers Dictionaries.
I. Microsoft Press computer dictionary.
QA76.15.M538 1999
004'.03--dc21 99-20168
CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 MLML 4 3 2 1 0 9

Distributed in Canada by ITP Nelson, a division of Thomson Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at mspress.microsoft.com.

Macintosh, Power Macintosh, QuickTime, and TrueType fonts are registered trademarks of Apple Computer, Inc. Kodak is a registered trademark of the Eastman Kodak Company. Intel is a registered trademark and Indeo is a trademark of Intel Corporation. Active Desktop, Active Directory, ActiveMovie, Active Platform, ActiveX, Authenticode, BackOffice, DirectInput, DirectX, Microsoft, Microsoft Press, MS-DOS, MSN, NetMeeting, NetShow, Visual Basic, Visual C++, Visual J++, WebTV, WebTV Network, Win32, Win32s, Windows, Windows NT, and XENIX are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. PANTONE is a registered trademark of Pantone, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person, or event is intended or should be inferred.

Acquisitions Editor: Christey Bahn
Project Editor: Kim Fryer

database *n.* A file composed of records, each containing fields together with a set of operations for searching, sorting, recombining, and other functions.

database administrator *n.* One who manages a database. The administrator determines the content, internal structure, and access strategy for a database, defines security and integrity, and monitors performance. *Acronym:* DBA. *Also called* database manager.

database analyst *n.* One who provides the analytic functions needed to design and maintain applications requiring a database.

database designer *n.* One who designs and implements functions required for applications that use a database.

database engine *n.* The program module or modules that provide access to a database management system (DBMS).

database machine *n.* **1.** A peripheral that executes database tasks, thereby relieving the main computer from performing them. **2.** A database server that performs only database tasks.

database management system *n.* A software interface between the database and the user. A database management system handles user requests for database actions and allows for control of security and data integrity requirements. *Acronym:* DBMS. *Also called* database manager. *See also* database engine.

database manager *n.* *See* database administrator, database management system.

database publishing *n.* The use of desktop publishing or Internet technology to produce reports containing information obtained from a database.

database server *n.* A network node, or station, dedicated to storing and providing access to a shared database. *Also called* database machine.

database structure *n.* A general description of the format of records in a database, including the number of fields, specifications regarding the type of data that can be entered in each field, and the field names used.

data bit *n.* In asynchronous communications, one of a group of from 5 to 8 bits that represents a single character of data for transmission. Data bits are preceded by a start bit and followed by an optional parity bit and one or more stop bits. *See also* asynchronous transmission, bit, communications parameter.

data buffer *n.* An area in memory where data is temporarily stored while being moved from one location to another. *See also* buffer[1].

data bus *n.* *See* bus.

data cable *n.* Fiber-optic or wire cable used to transfer data from one device to another.

data capture *n.* **1.** The collection of information at the time of a transaction. **2.** The process of saving on a storage medium a record of interchanges between a user and a remote information utility.

data carrier *n.* *See* carrier (definition 1).

Data Carrier Detected *n.* *See* DCD (definition 1).

data chaining *n.* The process of storing segments of data in noncontiguous locations while retaining the ability to reconnect them in the proper sequence.

data channel *n.* *See* channel (definition 1).

data collection *n.* **1.** The process of acquiring source documents or data. **2.** The grouping of data by means of classification, sorting, ordering, and other organizing methods.

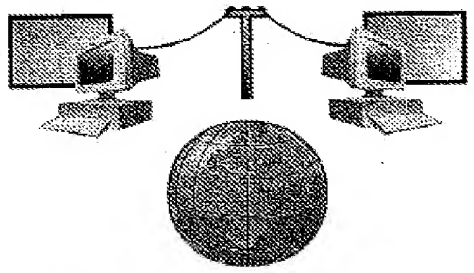
datacom *n.* Short for **data communications**. *See* communications.

data communications *n.* *See* communications.

data compaction *n.* *See* data compression.

data compression *n.* A means of reducing the amount of space or bandwidth needed to store or transmit a block of data, used in data communications, facsimile transmission, file storage and transfer, and CD-ROM publishing. *Also called* data compaction.

data conferencing *n.* Simultaneous data communication among geographically separated participants in a meeting. Data conferencing involves whiteboards and other software that enables a single set of files at one location to be accessed and modified by all participants. See the illustration. *See also* desktop conferencing, whiteboard. *Compare* video conferencing.



Data conferencing.

First-Name, Address, City, State, Zip-Code, Hire-Date, Current-Salary, Title, Department, and so on. Individual fields are characterized by their maximum length and the type of data (for example, alphabetic, numeric, or financial) that can be placed in them. The facility for creating these specifications usually is contained in the data definition language (DDL). In relational database management systems, fields are called *columns*. 2. A space in an on-screen form where the user can enter a specific item of information.

field-effect transistor *n.* *See* FET.

field-programmable logic array *n.* An integrated circuit containing an array of logic circuits in which the connections between the individual circuits, and thus the logic functions of the array, can be programmed after manufacture, typically at the time of installation in the field. Programming can be performed only once, typically by passing high current through fusible links on the chip. *Acronym:* FPLA. *Also called* PLA, programmable logic array.

field separator *n.* Any character that separates one field of data from another. *See also* delimiter, field (definition 1).

FIFO \'fī-fō\ *n.* *See* first in, first out.

fifth-generation computer *n.* *See* computer.

fifth normal form *n.* Abbreviated 5NF. *See* normal form (definition 1).

file *n.* A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another. A file is the "glue" that binds a conglomeration of instructions, numbers, words, or images into a coherent unit that a user can retrieve, change, delete, save, or send to an output device.

file allocation table *n.* A table or list maintained by some operating systems to manage disk space used for file storage. Files on a disk are stored, as space allows, in fixed-size groups of bytes (characters) rather than from beginning to end as contiguous strings of text or numbers. A single file can thus be scattered in pieces over many separate storage areas. A file allocation table maps available disk storage space so that it can mark flawed segments that should not be used and can find and link the pieces of a file. In MS-DOS, the file allocation table is commonly known as the FAT. *See also* FAT file system.

file attribute *n.* A restrictive label attached to a file that describes and regulates its use—for example, hidden, system, read-only, archive, and so forth. In MS-DOS, this information is stored as part of the file's directory entry.

file backup *n.* *See* backup.

file compression *n.* The process of reducing the size of a file for transmission or storage. *See also* data compression.

file control block *n.* A small block of memory temporarily assigned by a computer's operating system to hold information about an opened file. A file control block typically contains such information as the file's identification, its location on disk, and a pointer that marks the user's current (or last) position in the file. *Acronym:* FCB.

file conversion *n.* The process of transforming the data in a file from one format to another without altering the data—for example, converting a file from a word processor's format to its ASCII equivalent. In some cases, information about the data, such as formatting, may be lost. Another, more detailed, type of file conversion involves changing character coding from one standard to another, as in converting EBCDIC characters (which are used primarily with mainframe computers) to ASCII characters. *See also* ASCII, EBCDIC.

file extension *n.* *See* extension (definition 1).

file extent *n.* *See* extent.

file format *n.* The structure of a file that defines the way it is stored and laid out on the screen or in print. The format can be fairly simple and common, as are files stored as "plain" ASCII text, or it can be quite complex and include various types of control instructions and codes used by programs, printers, and other devices. Examples include RTF (Rich Text Format), DCA (Document Content Architecture), PICT, DIF (Data Interchange Format), DXF, TIFF (Tagged Image File Format), and EPSF (Encapsulated PostScript Format).

file fragmentation *n.* 1. The breaking apart of files as they are stored by the operating system into small, separate segments on disk. The condition is a natural consequence of enlarging files and saving them on a crowded disk that no longer contains contiguous blocks of free space large enough to hold them. File fragmentation is not an integrity problem, although it

several small squares displayed around a graphical object in a drawing program. The user can move or reshape the object by clicking on a handle and dragging. See the illustration.

Handle. A computer graphics handle.

handler *n.* 1. A routine that manages a common and relatively simple condition or operation, such as error recovery or data movement. 2. In some object-oriented programming languages that support messages, a subroutine that processes a particular message for a particular class of objects. *See also* message, object-oriented programming.

handshake *n.* A series of signals acknowledging that communication or the transfer of information can take place between computers or other devices. A hardware handshake is an exchange of signals over specific wires (other than the data wires), in which each device indicates its readiness to send or receive data. A software handshake consists of signals transmitted over the same wires used to transfer data, as in modem-to-modem communications over telephone lines.

hands-on *adj.* Involving interactive work with a computer or a computer program. A hands-on tutorial, for example, would teach a skill (such as the use of a program) by means of practice sessions and question-and-answer dialogues.

handwriting recognition *n.* 1. The ability of a computer to identify a user by recognizing features of handwriting, especially a signature. 2. The ability of a computer to translate handwritten text into character data for input. This technology is still under considerable development, and most handwriting recognition programs require users to form letters and words in a very consistent and clear manner to work adequately. The development of handwriting recognition programs has been spurred by PDAs, which frequently have keyboards that are too small for data entry, and software designed for Asian markets that have languages with numerous characters, which makes keyboards a cumbersome method for entering text. *See also* PDA. *Compare* optical character recognition.

hang *vb.* To stop responding. A hung program or computer system does not respond to user input, but the screen looks as if everything is running normally. The program or system might be waiting for something, for example, information from a network, or it might have terminated abnormally. It might resume running normally on its own, or the user might need to terminate and restart the program or reboot the computer. A hung computer system is said to be locked up. *See also* crash[1] (definition 1).

hanging indent *n.* Placement of the beginning of the first line of a paragraph farther to the left than the subsequent lines. *Also called* outdent. *Compare* indent.

hard *adj.* 1. Permanent, fixed, or physically defined; unchangeable by the ordinary operation of a computer system. *See also* hard copy, hard error, hard return, hard-sectored disk. *Compare* soft (definition 1). 2. Retaining magnetization even in the absence of an external magnetic field. *Compare* soft (definition 2).

hard card *n.* A circuit board, carrying a hard disk and containing its controller, that plugs into an expansion slot and uses the expansion bus for power as well as for data and control signals. By contrast, a hard disk in a drive bay communicates with a separate controller card by a ribbon cable and has a direct cable to the computer's main power supply. *See also* controller, drive bay, expansion slot, ribbon cable.

hard-coded *adj.* 1. Designed to handle a specific situation only. 2. Depending on values embedded in the program code rather than on values that can be input and changed by the user.

hard copy *n.* Printed output on paper, film, or other permanent medium. *Compare* soft copy.

hard disk *n.* A device containing one or more inflexible platters coated with material in which data can be recorded magnetically, together with their read/write heads, the head-positioning mechanism, and the spindle motor in a sealed case that protects against outside contaminants. The protected environment allows the head to fly 10 to 25 millionths of an inch above the surface of a platter rotating typically at 3,600 to 7,200 rpm; therefore, much more data can be stored and accessed much more quickly than on a floppy disk. Most hard disks contain from two to eight platters. See the illustration. *Also called* hard disk drive. *Compare* floppy disk.

hard disk drive *n. See* hard disk.

quartz crystal *n.* A precisely shaped and precisely sized piece of the mineral quartz, used for its piezoelectric properties. When a voltage is applied to a quartz crystal, it vibrates at a frequency determined by its size and shape. Quartz crystals are commonly used to control the frequency of oscillator circuits such as the clocks in microcomputers. *See also* piezoelectric.

quasi-language *n.* A derogatory term for any programming language that, because of deficiencies, is not suitable for any serious work.

query[1] *n.* A specific set of instructions for extracting particular data.

query[2] *vb.* To extract data from a database and present it for use.

query by example *n.* A simple-to-use query language implemented on several relational database management systems. Using query by example, the user specifies fields to be displayed, interable linkages, and retrieval criteria directly onto forms displayed on the screen. These forms are a direct pictorial representation of the table and row structures that make up the database. Thus, the construction of a query becomes a simple "checkoff" procedure from the viewpoint of the user. *Acronym:* QBE.

query language *n.* A subset of the data manipulation language, specifically, that portion relating to the retrieval and display of data from a database. It is sometimes used loosely to refer to the entire data manipulation language. *See also* data manipulation language.

Q

question mark *n. See* ?

queue \kyoo\ *n.* A multi-element data structure from which (by strict definition) elements can be removed only in the same order in which they were inserted; that is, it follows a first in, first out (FIFO) constraint. There are also several types of queues in which removal is based on factors other than order of insertion—for example, some priority value assigned to each element. *See also* deque, element (definition 1). *Compare* stack.

queued access method *n.* A programming technique that minimizes input/output delays by synchronizing the transfer of information between the program and the computer's input and output devices. *Acronym:* QAM.

QuickDraw *n.* On the Macintosh, the built-in group of routines within the operating system that control the display of graphics and text. Application programs call QuickDraw for on-screen displays. *See also* Toolbox.

QuickDraw 3-D *n.* A version of the Macintosh QuickDraw library that includes routines for doing 3-D graphics calculations. *See also* QuickDraw.

quicksort *n.* An efficient sort algorithm, described by C.A.R. Hoare in 1962, in which the essential strategy is to "divide and conquer." A quicksort begins by scanning the list to be sorted for a median value. This value, called the *pivot*, is then moved to its final position in the list. Next, all items in the list whose values are less than the pivot value are moved to one side of the list, and the items with values greater than the pivot value are moved to the other side. Each resulting side is sorted the same way, until a fully sorted list results. *See also* sort algorithm. *Compare* bubble sort, insertion sort, merge sort.

QuickTime *n.* Software components developed by Apple for creating, editing, publishing, and viewing multimedia content. QuickTime, which supports video, animation, graphics, 3D, VR (virtual reality), MIDI, music, sound, and text, has been part of the Mac OS since version 7 of the operating system and is used in many Macintosh applications. Windows applications can also run QuickTime files, but require the installation of special player software. QuickTime is often used on the Web to provide Web pages with video and animation. Most Web browsers support plug-ins for running these types of files. QuickTime is also part of the new MPEG-4 specification. *See also* MPEG-4.

Quick View *n.* A feature, optionally installed as part of Windows 9*x*, that provides a set of file viewers for previewing the contents of files without having to start the application(s) that created them. The feature is accessed through the Quick View command, available either from the File menu or by right-clicking a filename. If the feature has been installed but the file type is not supported by a viewer, the Quick View command does not appear.

quit[1] *n.* 1. An FTP command that instructs the server to drop the current connection with the client from which it received the command. 2. A command in many applications for exiting the program.

XI. RELATED PROCEEDINGS APPENDIX

Not applicable in the present appeal.